A

awsui

Verified AWS contexts for AI agents and CLI tools.

About

awsui is a verified AWS context launcher for AI agents and CLI workflows. It lets you pick dev, qa, staging, or prod once, checks the real AWS identity with STS, and starts the command only if the account matches the project's expected environment. That makes it useful for Codex, Claude Code, Terraform, and other tools that should run in the right AWS account without relying on a local profile name as proof.

What is awsui

awsui is a command-line tool for running AI agents and other CLIs inside a verified AWS account context. It is built around the idea that an environment name such as dev, qa

staging, or prod should map to an expected AWS account and region, and that mapping should be checked before anything else starts.

The project positions itself as a guardrail for teams that work across multiple AWS accounts. Instead of trusting a developer's current profile label, awsui authenticates first, checks the actual STS identity, and then launches the requested child process only when the account matches the declared project environment.

In practice, that means you can use the same environment intent across Codex, Claude Code, Kiro, Terraform, Antigravity, scripts, or a shell session without sharing each developer's local profile names.

Key features

- **STS verification before launch:** awsui checks the authenticated AWS identity before it starts the command.
- **Fail-closed behavior:** if the account is wrong or the identity check does not pass, the child process never begins.
- **Project-level environment mapping:** you define environments such as prod once in a shared `.awsui.yaml` file.
- **Local profile binding:** each developer maps those shared environments to their own machine-specific AWS profile names outside Git.
- **Region handling:** the tool sets the intended region as part of the verified launch flow.
- **Tool-agnostic workflow:** it can launch AI coding assistants, infrastructure commands, or any other CLI.
- **Credential hygiene:** direct credential overrides are removed from the launch environment, and awsui does not store AWS credentials itself.

How to use it

- **Install the tool.** The site shows installation with `uv tool install --python 3.13 awsui`, which suggests a Python-based CLI setup.
- **Initialize the project.** Run `awsui init` inside your AWS-backed project to create the shared context definitions.

Bind your local profile. Use `awsui bind prod` or a similar command to connect that shared environment to your personal AWS profile name.

Verify and launch. Run `awsui prod -- codex` or another command to authenticate, confirm the STS account, and start the tool only after the check passes.

Use the same flow for other commands. The site also shows forms like `awsui context prod` and `awsui run prod -- ...`, indicating support for explicit verified launches.

Who it is for

awsui is a fit for teams and individuals who work in multiple AWS accounts and want stronger launch-time safety.

AI agent users: developers running Codex or Claude Code in account-specific environments.

Platform and DevOps teams: people who need Terraform and deployment CLIs to run against the intended account every time.

Multi-account AWS teams: groups that want shared environment intent without committing personal profile names.

Security-conscious workflows: anyone who wants an identity check to happen before a process gets control of credentials or environment setup.

Developers switching often between accounts: users who need a deterministic way to avoid mixing up prod, staging, and dev.

What to know before you use it

awsui is designed as a guardrail, not as a replacement for AWS IAM or account design. It verifies the authenticated STS identity against the expected account before launch, but it does not itself manage AWS permissions or create AWS accounts.

A few practical limits are worth noting:

It relies on AWS credentials already managed by the AWS CLI; it is not shown as a standalone credential store.

The site describes project setup around `.awsui.yaml`, but the exact file schema is not fully documented in the harvested text.

If you need authoritative account, region, or permission details, use AWS's own console, IAM, or STS sources as the source of truth.

The source material does not confirm GUI support, hosting as a web app, or mobile platforms, so treat awsui as a terminal-first developer tool.

If your workflow depends on the right AWS account being verified before a CLI starts, awsui adds that checkpoint without requiring you to trust a profile name alone.

FAQ

What does awsui do in one sentence?

Which tools can it launch?

How does awsui prevent the wrong account from being used?

Does awsui store AWS credentials?

Is awsui meant for teams or solo developers?

Sponsored

Belay – Awa... Featured
Developer Tools

Stays awake while your agents work.

Free # Developer Tools

View ↗

Partly Featured
Finance

Partly is a mobile app that lets friends quickly scan receipts, split any bill fairl...

Free # Finance

View ↗

awsui

Pricing	Free
Category	AI
Platform	Web, API
Launched	Aug 2026
Updated	19 days ago
Verified	20 days ago

TAGS

ai-agents # aws # devops # sts # terraform # cli # security # multi-account

Dir Hub

A curated directory of small, well-made apps. Browse by category or search to find the right one in fewer clicks.

DISCOVER

Home

Pricing

Submit your app

COMPANY

About

Contact

LEGAL

Privacy

Terms

© 2026 Dir Hub

Curated indie apps

A awsui